R Research Project on Weather

Team Composition:

Pankevych Yevhen, Butynets Danylo

Aim of the Project

We process the data of weather of certain town and try to build a model that detemines wheather the Humidity will be high based on the visibility, temperature and wind speed. Also, we try to determine the distribution of the Humidity.

The data is taken from this site - https://www.kaggle.com/rtatman/datasets-for-regression-analysis. (2nd dataset)

Reading the Data

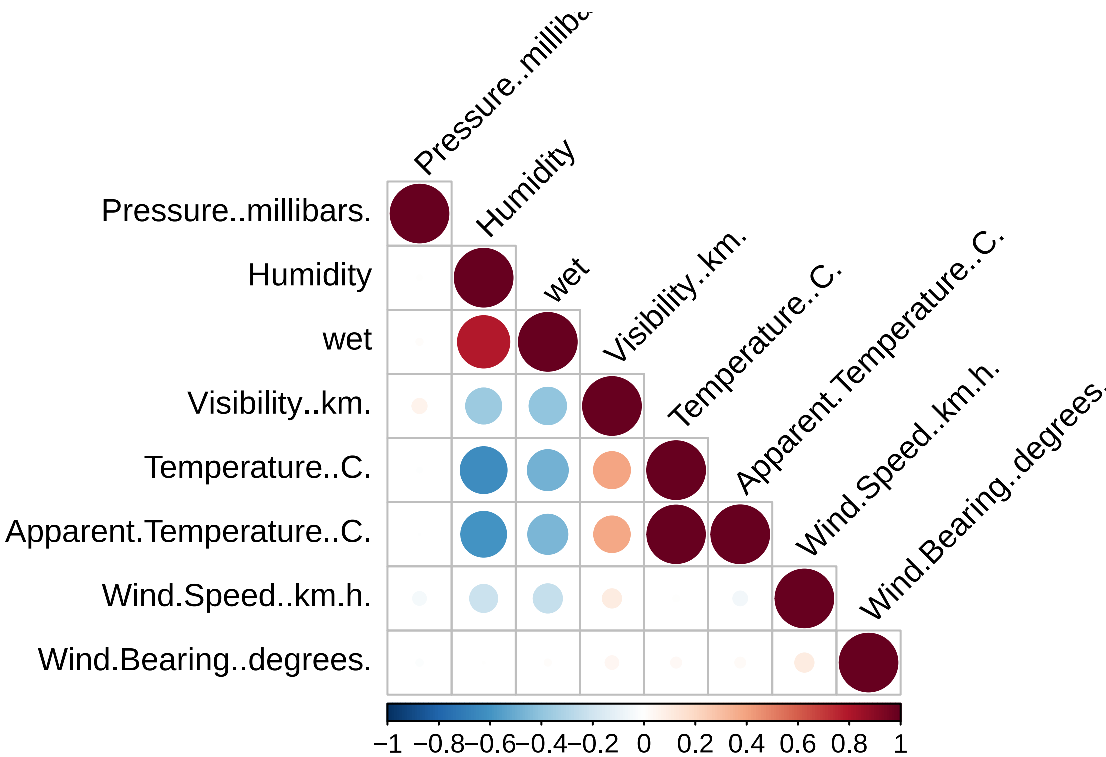
```
weather = read.csv("weatherHistory.csv")

# clean/remove some unnecessary text data
weather[1] = NULL
weather[1] = NULL
weather[1] = NULL
weather[9] = NULL
weather[7] = NULL

# For convinience, add one collum "Wet" that will take True or False
weather$wet = weather$Humidity > 0.8
summary(weather)
```

```
##  Temperature..C.   Apparent.Temperature..C.    Humidity      Wind.Speed..km.h.
##  Min.   :-21.822   Min.   :-27.717          Min.   :0.0000   Min.   : 0.000
##  1st Qu.:  4.689   1st Qu.:  2.311          1st Qu.:0.6000   1st Qu.: 5.828
##  Median : 12.000   Median : 12.000          Median :0.7800   Median : 9.966
##  Mean   : 11.933   Mean   : 10.855          Mean   :0.7349   Mean   :10.811
##  3rd Qu.: 18.839   3rd Qu.: 18.839          3rd Qu.:0.8900   3rd Qu.:14.136
##  Max.   : 39.906   Max.   : 39.344          Max.   :1.0000   Max.   :63.853
##  Wind.Bearing..degrees. Visibility..km. Pressure..millibars.    wet
##  Min.   :  0.0          Min.   : 0.00   Min.   :   0        Mode :logical
##  1st Qu.:116.0          1st Qu.: 8.34   1st Qu.:1012        FALSE:52192
##  Median :180.0          Median :10.05   Median :1016        TRUE :44261
##  Mean   :187.5          Mean   :10.35   Mean   :1003
##  3rd Qu.:290.0          3rd Qu.:14.81   3rd Qu.:1021
##  Max.   :359.0          Max.   :16.10   Max.   :1046
```

Distribution of parameters based on Wet parameter and Best correlation

Start with linear regression Find some correlations in data

```r
rquery.cormat(weather)
```

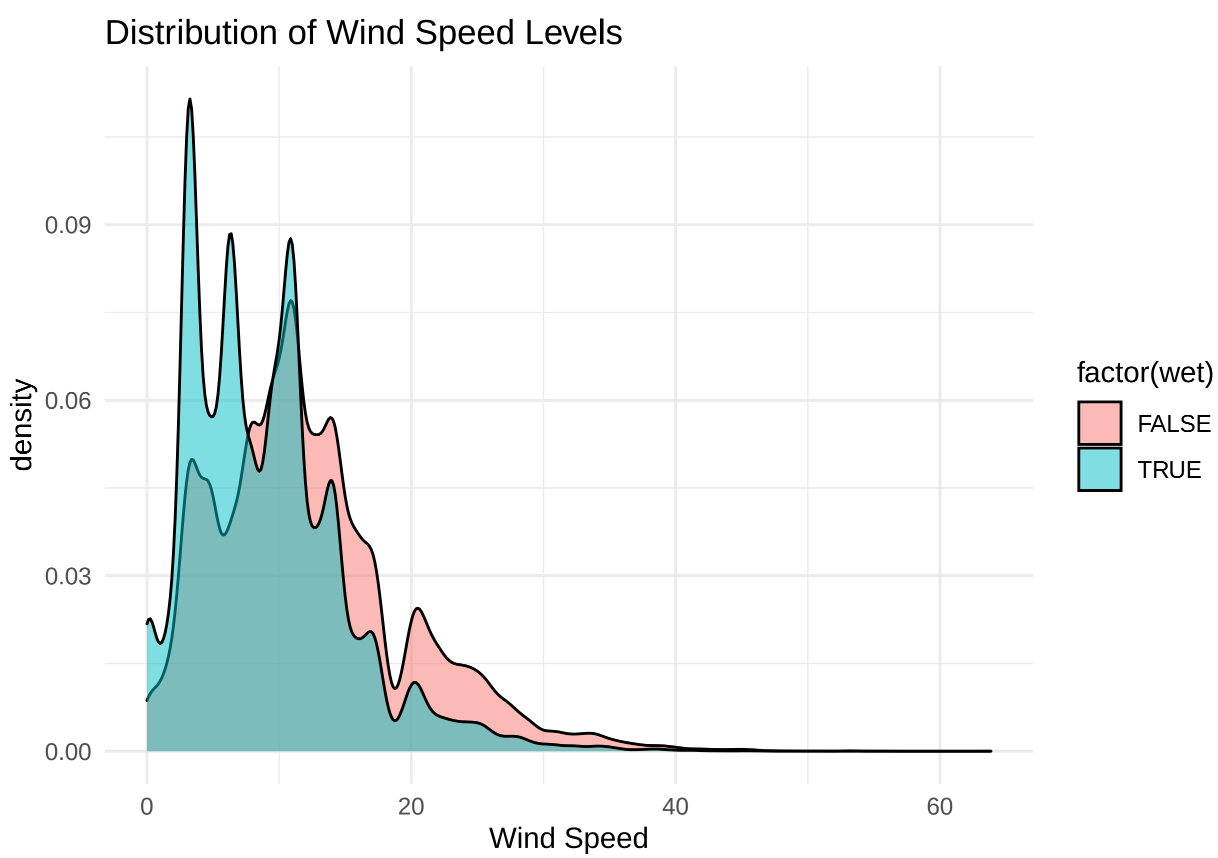



```
## $r
##                          Pressure..millibars. Humidity   wet Visibility..km.
## Pressure..millibars.                        1
## Humidity                               0.0055        1
## wet                                     0.011     0.79     1
## Visibility..km.                          0.06    -0.37  -0.4               1
## Temperature..C.                       -0.0054    -0.63 -0.48            0.39
## Apparent.Temperature..C.             -0.00022     -0.6 -0.46            0.38
## Wind.Speed..km.h.                      -0.049    -0.22 -0.24             0.1
## Wind.Bearing..degrees.                 -0.012  0.00073 0.011           0.048
##                          Temperature..C. Apparent.Temperature..C.
## Pressure..millibars.
## Humidity
## wet
## Visibility..km.
## Temperature..C.                        1
## Apparent.Temperature..C.            0.99                        1
## Wind.Speed..km.h.                  0.009                   -0.057
## Wind.Bearing..degrees.              0.03                    0.029
```

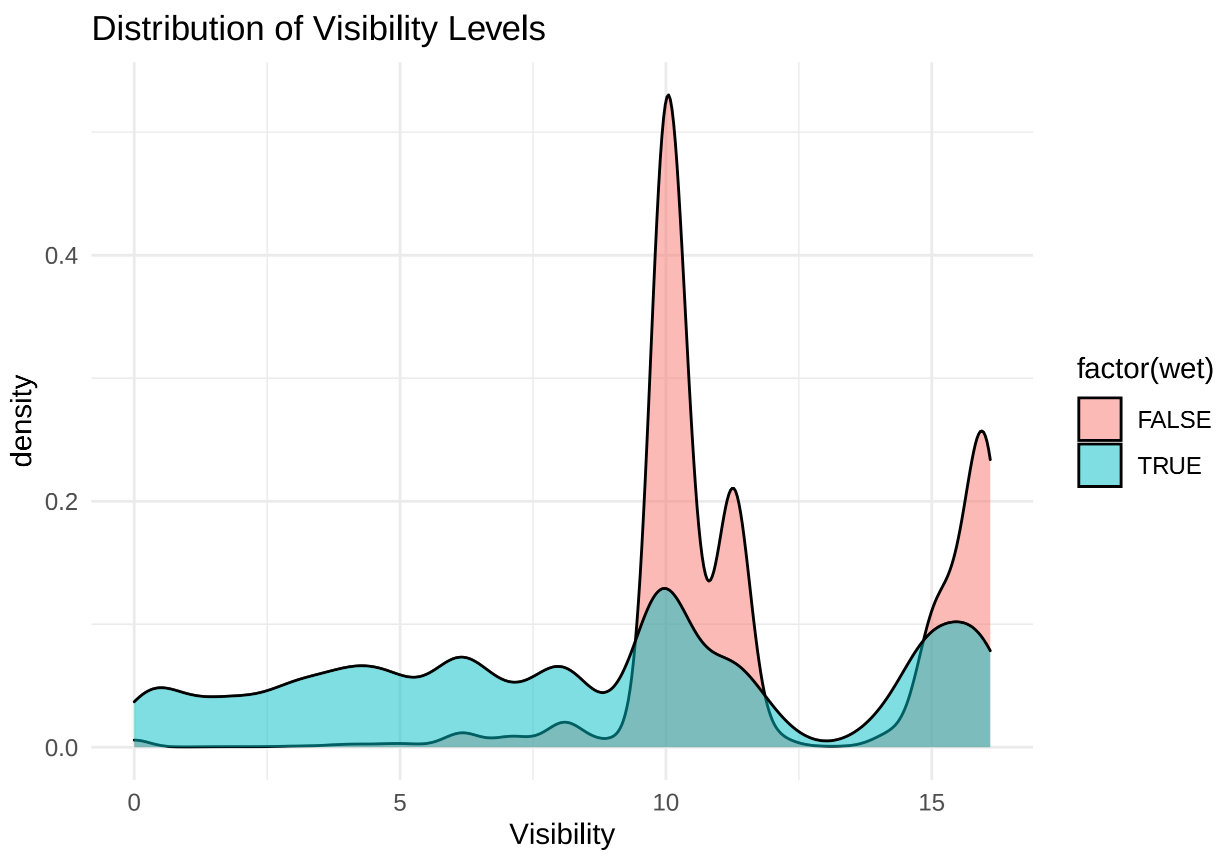
```
##                          Wind.Speed..km.h. Wind.Bearing..degrees.
## Pressure..millibars.
## Humidity
## wet
## Visibility..km.
## Temperature..C.
## Apparent.Temperature..C.
## Wind.Speed..km.h.                        1
## Wind.Bearing..degrees.                 0.1                      1
##
## $p
##                          Pressure..millibars. Humidity     wet Visibility..km.
## Pressure..millibars.                        0
## Humidity                                 0.09        0
## wet                                     6e-04        0       0
## Visibility..km.                       3.6e-77        0       0               0
## Temperature..C.                         0.091        0       0               0
## Apparent.Temperature..C.                 0.95        0       0               0
## Wind.Speed..km.h.                     6.7e-53        0       0       5.4e-216
## Wind.Bearing..degrees.                  3e-04     0.82 0.00083        1.7e-49
##                          Temperature..C. Apparent.Temperature..C.
## Pressure..millibars.
## Humidity
## wet
## Visibility..km.
## Temperature..C.                        0
## Apparent.Temperature..C.               0                        0
## Wind.Speed..km.h.                 0.0054                  2.2e-69
## Wind.Bearing..degrees.           1.2e-20                  1.9e-19
##                          Wind.Speed..km.h. Wind.Bearing..degrees.
## Pressure..millibars.
## Humidity
## wet
## Visibility..km.
## Temperature..C.
## Apparent.Temperature..C.
## Wind.Speed..km.h.                        0
## Wind.Bearing..degrees.           2.6e-229                      0
##
## $sym
##                          Pressure..millibars. Humidity wet Visibility..km.
## Pressure..millibars.     1
## Humidity                                      1
## wet                                           ,        1
## Visibility..km.                               .        .   1
## Temperature..C.                               ,        .   .
## Apparent.Temperature..C.                      .        .   .
## Wind.Speed..km.h.
## Wind.Bearing..degrees.
##                          Temperature..C. Apparent.Temperature..C.
## Pressure..millibars.
## Humidity
## wet
## Visibility..km.
```

```
## Temperature..C.            1
## Apparent.Temperature..C. B                1
## Wind.Speed..km.h.
## Wind.Bearing..degrees.
##                          Wind.Speed..km.h. Wind.Bearing..degrees.
## Pressure..millibars.
## Humidity
## wet
## Visibility..km.
## Temperature..C.
## Apparent.Temperature..C.
## Wind.Speed..km.h.        1
## Wind.Bearing..degrees.                    1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

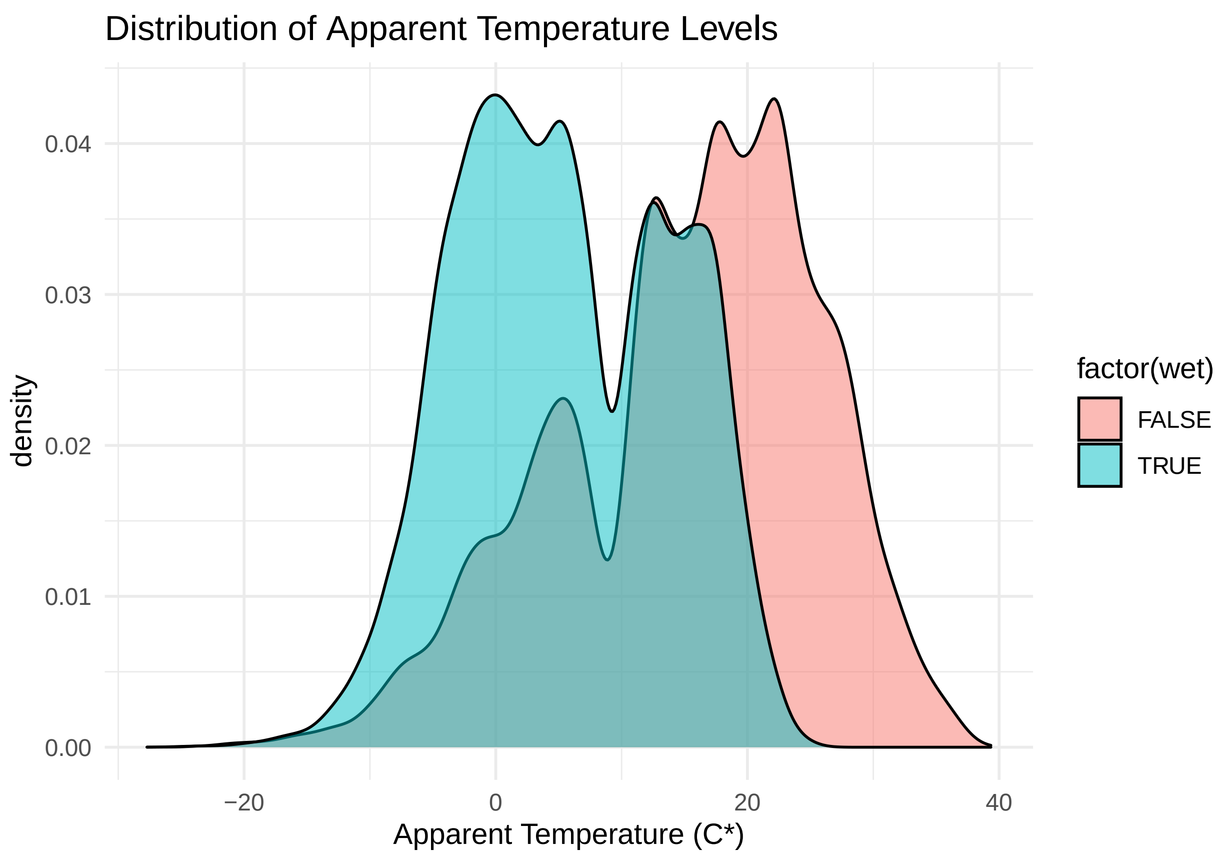
```r
ggplot(weather,aes(x=Wind.Speed..km.h.,fill=factor(wet)))+geom_density(alpha=0.5)+
 xlab(label = "Wind Speed")+
 ggtitle("Distribution of Wind Speed Levels")+
 theme_minimal()
```



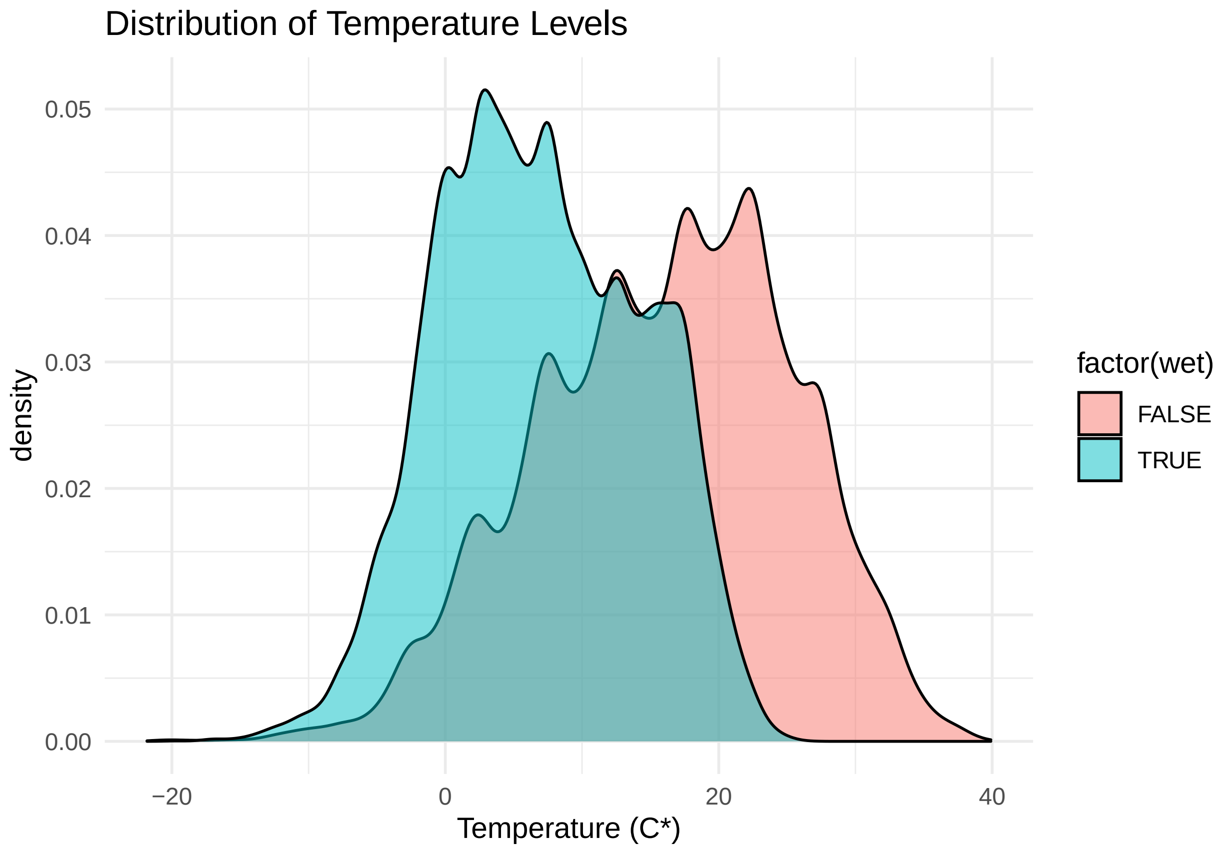

```r
ggplot(weather,aes(x=Visibility..km.,fill=factor(wet)))+geom_density(alpha=0.5)+
 xlab(label = "Visibility")+
 ggtitle("Distribution of Visibility Levels")+
 theme_minimal()
```




```
ggplot(weather,aes(x=Apparent.Temperature..C.,fill=factor(wet)))+geom_density(alpha=0.5)+
 xlab(label = "Apparent Temperature (C*)")+
 ggtitle("Distribution of Apparent Temperature Levels")+
 theme_minimal()
```




```r
ggplot(weather,aes(x=Temperature..C.,fill=factor(wet)))+geom_density(alpha=0.5)+
 xlab(label = "Temperature (C*)")+
 ggtitle("Distribution of Temperature Levels")+
 theme_minimal()
```



We can see correlation between Humidity and Temperature and Apparent. temperature, and also small correlation between Humidity and Visibility + Wind speed. But for our convinience we will use Wet instead of Humidity. Lets try calculating following and finding the best correlation:

1. wet ~ Temperature + Apparent. temperature + Visibility + Wind speed
2. wet ~ Temperature + Apparent. temperature + Visibility
3. wet ~ Temperature + Apparent. temperature
4. wet ~ Temperature

```
summary(lm(wet~Temperature..C. + Apparent.Temperature..C. + Visibility..km. +
             Wind.Speed..km.h., data=weather))$r.squared
```

```
## [1] 0.3279943
```

```
summary(lm(wet~Temperature..C. + Apparent.Temperature..C. + Visibility..km.,
           data=weather))$r.squared
```

```
## [1] 0.2998584
```

```
summary(lm(wet~Temperature..C. + Apparent.Temperature..C.,
           data=weather))$r.squared
```

```
## [1] 0.2499833
```

```r
summary(lm(wet~Temperature..C., data=weather))$r.squared
```

```
## [1] 0.2279855
```

The best correlation for linear regression we got for all 4 features, so we can use them.

Training the model

Now divide dataset to train and test data

```r
#weather$wet <- as.factor(weather$wet)
train <- weather[1:85000, ]
test <- weather[85001:96453,]
```

In fact, we will generate coefficients α_1 $\alpha_2, \alpha_3, \alpha_4$ for each feature (Temperature + Apparent. temperature + Visibility + Wind speed) Here we used dictret boolean values for "Wet" characteristic. Our model will be a simple classifier that by given observation of features (mentioned upper) will generate a classification, can this given observation belong to the "Wet" category. When predicting, our model will return a probability $P(wet = True|x_1, x_2, x_3, x_4)$, where x_n - one feature. So we will take observation to be "Wet" if this probability >= 0.5

```r
model <- lm(wet ~ Temperature..C. + Apparent.Temperature..C. +
              Visibility..km. + Wind.Speed..km.h., data = train)
summary(model)
```

```
##
## Call:
## lm(formula = wet ~ Temperature..C. + Apparent.Temperature..C. +
##     Visibility..km. + Wind.Speed..km.h., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5221 -0.3095 -0.0747  0.3442  1.1276
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.1497721  0.0044901 256.071  < 2e-16 ***
## Temperature..C.          -0.0305008  0.0014201 -21.479  < 2e-16 ***
## Apparent.Temperature..C.  0.0096669  0.0012691   7.617 2.63e-14 ***
## Visibility..km.          -0.0278653  0.0003706 -75.189  < 2e-16 ***
## Wind.Speed..km.h.        -0.0143542  0.0002410 -59.560  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4081 on 84995 degrees of freedom
## Multiple R-squared:  0.3272, Adjusted R-squared:  0.3272
## F-statistic: 1.033e+04 on 4 and 84995 DF,  p-value: < 2.2e-16
```

```r
model$coefficients
```

```
##              (Intercept)          Temperature..C. Apparent.Temperature..C.
##              1.149772098             -0.030500845              0.009666923
##            Visibility..km.        Wind.Speed..km.h.
##             -0.027865348             -0.014354240
```

And now we try to predict if it will be wet today:

```
prediction <- predict.lm(model, newdata = test, type = 'response')
prediction <- ifelse(prediction >= 0.5,TRUE,FALSE)
result <- data.frame(prediction)

# Take as factor only because we want to use confusionMatrix, which will calculate an accuracy
result$prediction <- as.factor(result$prediction)
test$wet <- as.factor(test$wet)

confusionMatrix(result$prediction, test$wet)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  4707 2322
##      TRUE    718 3706
##
##                Accuracy : 0.7346
##                  95% CI : (0.7264, 0.7426)
##     No Information Rate : 0.5263
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4754
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8676
##             Specificity : 0.6148
##          Pos Pred Value : 0.6697
##          Neg Pred Value : 0.8377
##              Prevalence : 0.4737
##          Detection Rate : 0.4110
##    Detection Prevalence : 0.6137
##       Balanced Accuracy : 0.7412
##
##        'Positive' Class : FALSE
##
```

Accuracy was calculated as $ACC = \frac{TruePos+TrueNeg}{TestLength}$

As we saw, it gives quite good results (73% of accuracy), and it beat No Information Rate - we created a model that can be used in real life!

Now let's try finding some another linear dependency! As we saw on the correlation plot, we got some correlation between Visibility and (Hymidity, Temperature and Apparent. temp. and Wind.Speed.). Let's try predicting Visibility!

```r
summary(lm(Visibility..km. ~ Humidity + Temperature..C. +
             Apparent.Temperature..C. + Wind.Speed..km.h., data=weather))$r.squared
```

```
## [1] 0.1817025
```

R square is quite small, but let's try

```r
model <- lm(Visibility..km. ~ Humidity + Temperature..C. +
              Apparent.Temperature..C. + Wind.Speed..km.h., data = train)
summary(model)
```

```
##
## Call:
## lm(formula = Visibility..km. ~ Humidity + Temperature..C. + Apparent.Temperature..C. +
##     Wind.Speed..km.h., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7755  -2.6130  -0.3701   2.8106   9.0562
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              10.988301   0.092014 119.420  < 2e-16 ***
## Humidity                 -3.472363   0.088523 -39.226  < 2e-16 ***
## Temperature..C.           0.105271   0.013322   7.902 2.78e-15 ***
## Apparent.Temperature..C.  0.011354   0.011747   0.967    0.334
## Wind.Speed..km.h.         0.040140   0.002237  17.944  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.743 on 84995 degrees of freedom
## Multiple R-squared:  0.1721, Adjusted R-squared:  0.1721
## F-statistic:  4418 on 4 and 84995 DF,  p-value: < 2.2e-16
```

In this case we created a model that do not classify, but rather try to estimate numeric value. So we need to find how big is correlation between test data and predicted.

```r
prediction <- predict(model, test)
result <- data.frame(prediction)

actuals_preds <- data.frame(cbind(actuals=test$Visibility..km., predicteds=result$prediction))
correlation_accuracy <- cor(actuals_preds)

correlation_accuracy
```

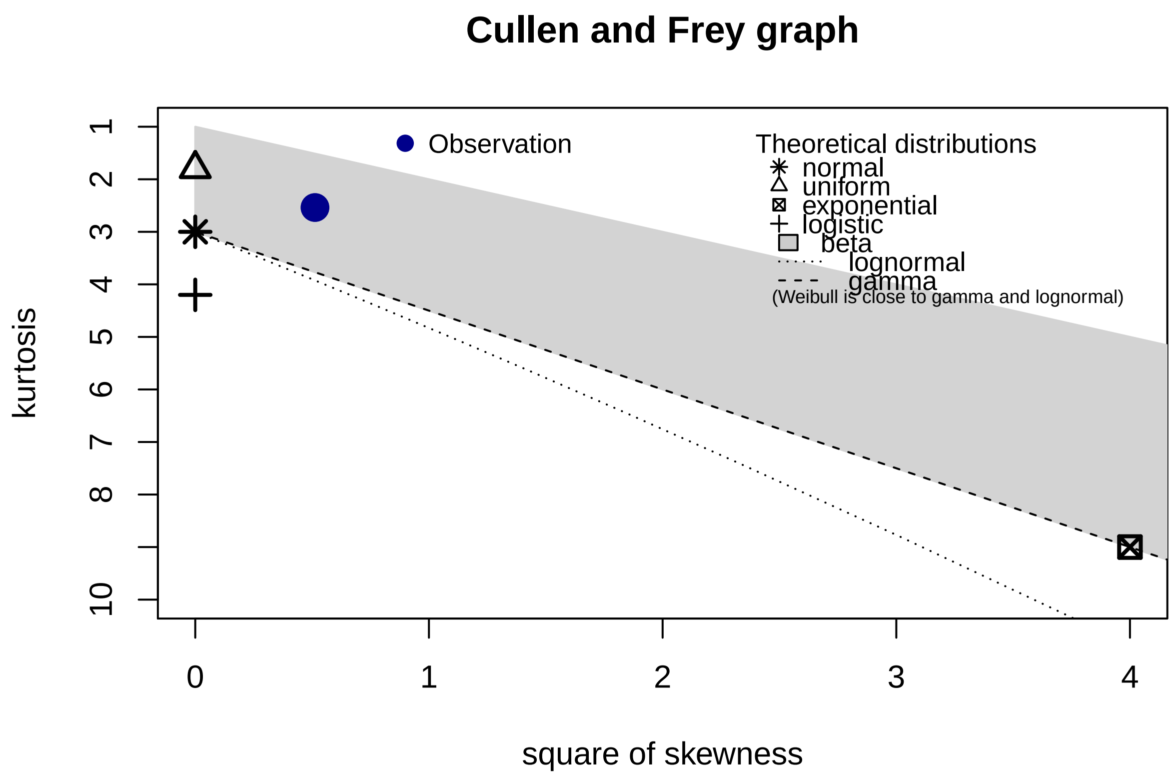
```
##             actuals predicteds
## actuals    1.000000   0.530624
## predicteds 0.530624   1.000000
```

This time accuracy is not so good (about 50%). There are some reasons for it. Firstly this time we haven't discritised our values (we used real numbers, but not TRUE/FALSE). Also we got small R squared value, so there was not so good correlation. But bad results are results too, so we understood that knowing humidity, temperatures and wind speed is not enough to predict visibility.

Finding the distribution of Humidity

Now let's perform some tests to discover a distribution of Humidity

```r
descdist(weather$Humidity)
```

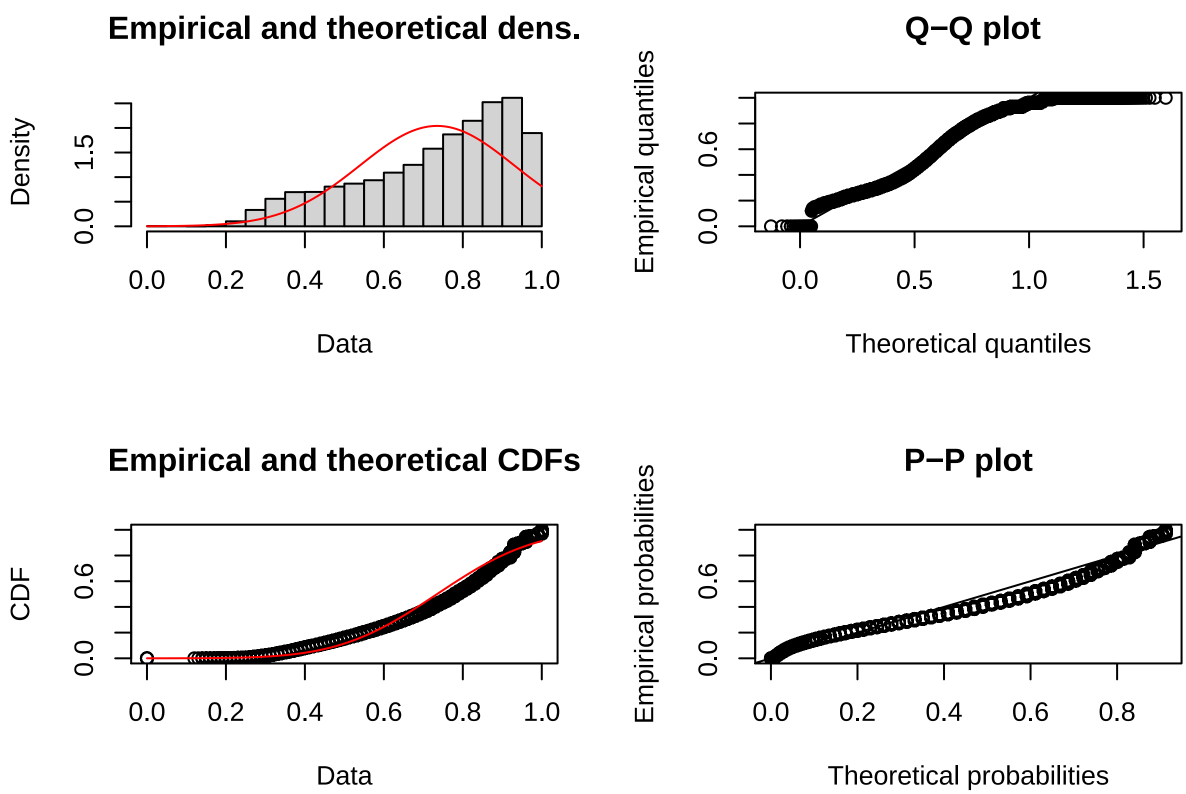



```
## summary statistics
## ------
## min:  0   max:  1
## median:  0.78
## mean:  0.734899
## estimated sd:  0.1954727
## estimated skewness:  -0.7158804
## estimated kurtosis:  2.53783
```

As we see, our data's skewness and kutorsis are between normal and uniform distributions. Let's try to fit them both

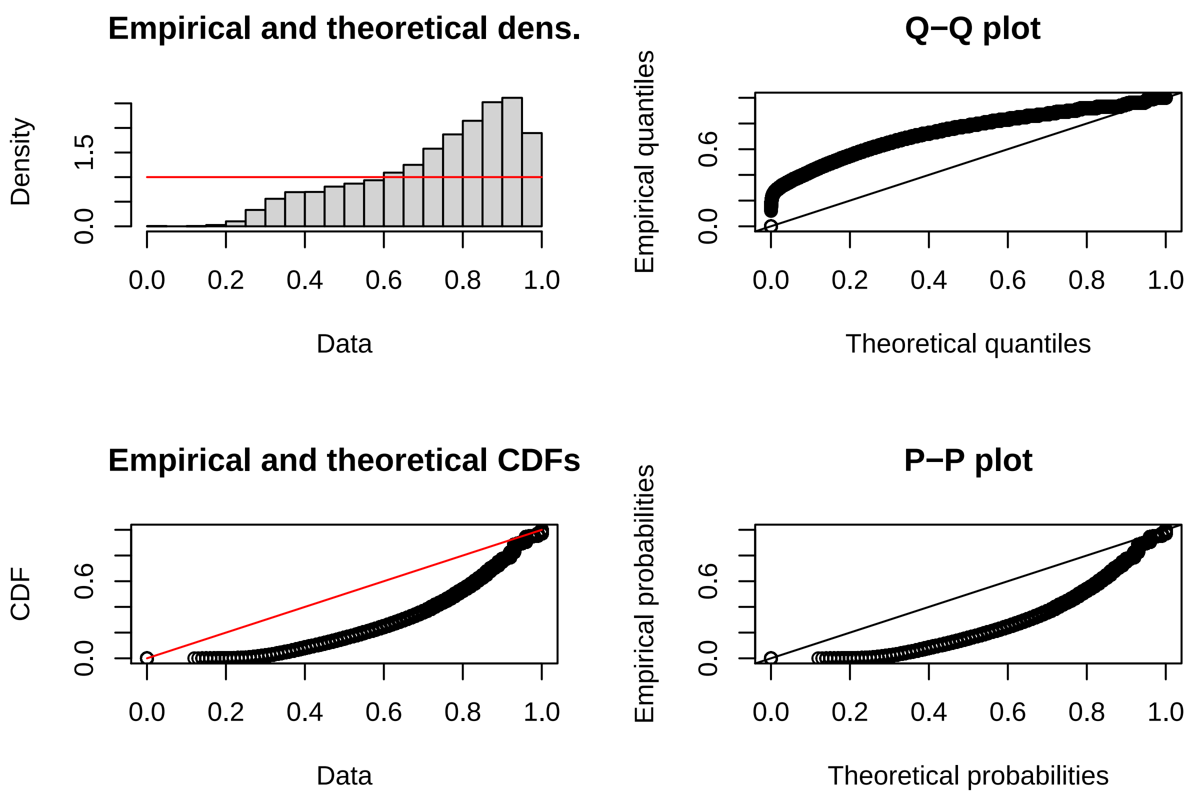
First about normal distribution

```r
humidity.fit.norm = fitdist(weather$Humidity, "norm")
plot(humidity.fit.norm)
```



And now about uniform

```
humidity.fit.unif = fitdist(weather$Humidity, "unif")
plot(humidity.fit.unif)
```



Now compare two aic values

```
humidity.fit.norm$aic
```

```
## [1] -41162.33
```

```
humidity.fit.unif$aic
```

```
## [1] 4
```

As we see, data is likely to be normally distributed Now perform Kolmogorov-Smirnov test for both normal and uniform distributions, just to make sure that it is more likely for data to be normally distributed. We will perform 2 tests, with H_0 that will be "data is normally/uniformly distributed" (or in terms of Kolmogorov-Smirnov that data's ecdf is close enough to the cdf of corresponding distribution) and H_1 that "data is not normally/uniformly distributed".

```
ks.test(weather$Humidity, "pnorm", mean(weather$Humidity), sd(weather$Humidity))
```

```
## Warning in ks.test(weather$Humidity, "pnorm", mean(weather$Humidity),
## sd(weather$Humidity)): ties should not be present for the Kolmogorov-Smirnov
## test
```

```
##
##  One-sample Kolmogorov-Smirnov test
```

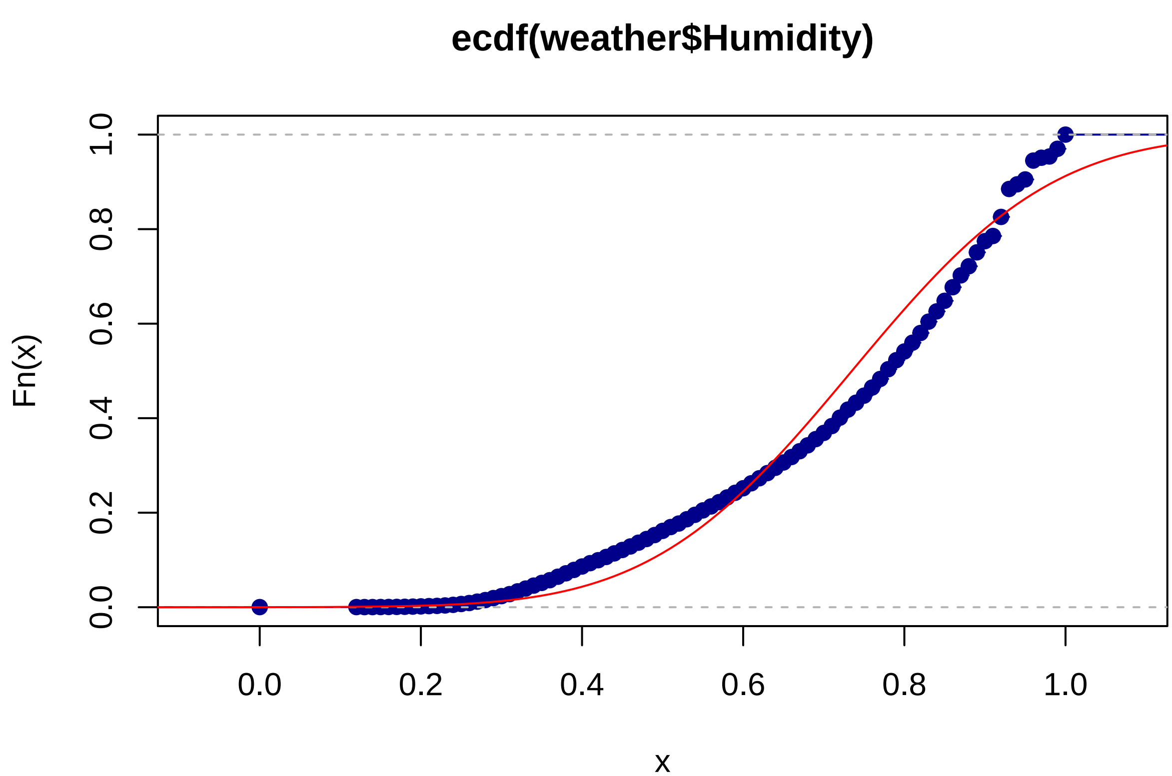
```
##
## data:  weather$Humidity
## D = 0.10868, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
ks.test(weather$Humidity, "punif")
```

```
## Warning in ks.test(weather$Humidity, "punif"): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  weather$Humidity
## D = 0.3583, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

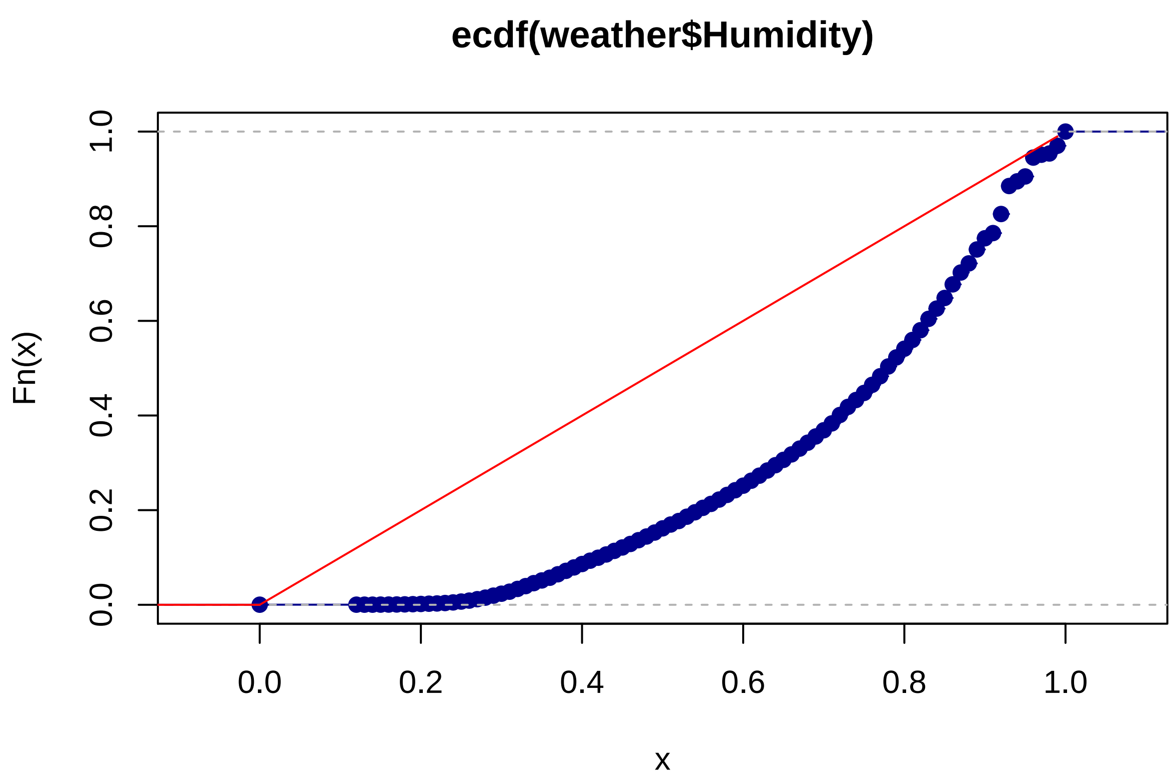
```r
x <- rnorm(length(weather$Humidity), mean=mean(weather$Humidity), sd=sd(weather$Humidity))
pts <- seq(-1,max(x),by=0.01)
plot(ecdf(weather$Humidity),col="darkblue")
lines(pts, pnorm(pts, mean=mean(weather$Humidity), sd=sd(weather$Humidity)), col="red")
```



```r
maxDiff = max(pnorm(pts, mean=mean(weather$Humidity),
                    sd=sd(weather$Humidity))-ecdf(weather$Humidity)(pts))
cat("Maximal difference between ecdf and cdf: ", maxDiff, "\n")
```

```
## Maximal difference between ecdf and cdf:  0.08991411
```

```r
x <- runif(length(weather$Humidity))
pts <- seq(-1,max(x),by=0.01)
plot(ecdf(weather$Humidity),col="darkblue")
lines(pts, punif(pts), col="red")
```




```r
maxDiff = max(punif(pts) - ecdf(weather$Humidity)(pts))
cat("Maximal difference between ecdf and cdf: ", maxDiff, "\n")
```

```
## Maximal difference between ecdf and cdf:  0.3522183
```

However D value (absolute max distance between the CDFs of the two samples, which is calculated as $D_n = sup_x|F_n(x) - F(x)|$ where $F_n(x)$ is ecdf from data and $F(x)$ is cdf of concrete distribution) of first test is quite small (0.1) and smaller than from the second test, but as p value is almost 0, we can't say that data is totally normally distributed. However it's ecdf is close to the normal cdf, so we can assume that this data's distribution is quite close to normal one.

Conclusions

We tried to use not a single approach to the data, and tried to make to models instead of one. The second one didn't result to be accurate, but despite failing here, we analyzed data from multiple perspectives. Despite that, the first model we made is having pretty high accuracy and we also managed to determine the distribution of the Humidity. Overall, we consider this project as a successfil one.